# PATTERN MATCHING (Tasarım eşleştirmeleri)

Akış kontrol mekanizmalarında yapabildiğimiz şeylerin daha da desenleşmiş halidir. Olayı daha da kolaylaştırmamızı sağlar.

- C# 7.0 ile
  - ▼ Type matching
    - Object içerisindeki bir tipin belirlenmesinde kullanılan "is" operatörünün desenleştirilmiş halidir.
    - is ile belirlenen türün direkt dönüşümünü sağlar

```
object x = 125;
if (x is string)
    string xx = x as string;
    Console.WriteLine($"x değişkeni string tipindedir.");
else if (x is int)
    int xx = (int)x;
    Console.WriteLine($"x değişkeni int tipindedir.");
object x = 125;
if (x is string xx)
   Console.WriteLine($"x değişkeni string tipindedir.");
else if (x is int xx)
   Console.WriteLine($"x değişkeni int_tipindedir_"):
```

object x= "irem";
if (x is string a)
{

Console.Writeline(a); // a ya buradan erişemeyiz.

// Çünkü; type pattern da tanımlanan değişkenlere manuel orangan değişkenl

// null olan değişkenler herhangi bir yerden çağırılamaz. //a nın nuull olma ihtimalinin sebebi ise, x ya string de

•

#### ▼ Constant Pattern

 Elimizdeki veriyi sabit bir değer ile karşılaştırabilmemizi sağlar. == kontrolünü sağlar kısaca.

•

```
object x=123; //kontrol edeceğimiz değer object olmak zoru
if(x is 123) //değer bazlı bir kontroldür. Eğer değer kont
{
}

if (x is int) //tür bazlı bir kontroldür. Eğer tür kontrol
{
}
```

•

### ▼ var Pattern

- Eldeki veriyi var değişkeni ile elde etmemizi sağlamaktadır.
- ÇOK ÖNEMLİ: Normalde kullanılan var derleyici sürecinde türü belirlerken, burada kullanılan var türü runtime da belirler.

```
object x ="asdfg";
if(x is var a) // x in türü her ne ise a ya unboxing enede
{
}
```

#### ▼ Recursive Pattern

• Switch bloğunda referans türlü değişkenlerde kontrol edilmektedir.

 Case null komutu ile ilgili türün/ referansın null olup olmamasını kontrol edebilmeisnden dolayı Constant pattern'ı kapsamaktadır.

•

```
object x="lkdnclkn";
if(x is string a)
{
Console.Write (a);
}

if (x is var b)
{
}

bool result = x is string o1;
Console.Writeline (o1); // Hata verir. Type pattern 'da x

bool result2 = x is string o2;
Console.Writeline (o2); // var pattern'da ise değişkenin ox
```

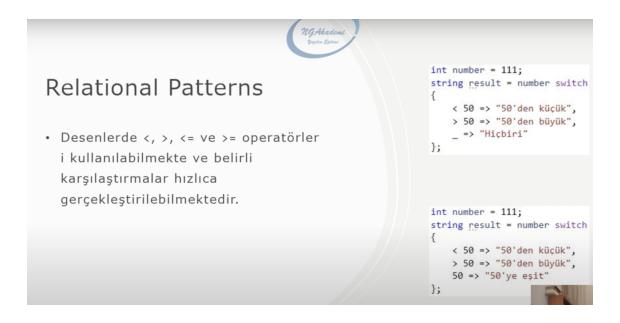
•

- C# 9.0ile
  - ▼ Simple type pattern

Değişken isimleri zorunlulupunu kaldırır.

```
object obj = new Person();
switch (obj)
                                                              Simple Type
     case Person p:
           //...
                                                              Patterns
           break;
                                                               · Bir değişken içerisindeki değerin belirli bir
                                                                  türde olup olmadığını hızlı bir şekilde
object obj = new Person();
                                                                  kontrol etmemizi sağlayan desendir.
switch (obj)
                                                               • C# 9.0'dan önce Type Pattern ile yapılan
                                                                  tür bildirimlerinin yanına değişken adı
                                                                  tanımlaması yahut discard ifadesinin
     case Person:
                                                                  kullanılması zaruriydi. C# 9.0 ile bu
                                                                  gereksiz zorunluluk ortadan kaldırılmış ve
          //...
                                                                  direkt olarak tür kontrol islemine
           break;
                                                                  odaklanılması sağlanmıştı
```

#### ▼ Relations Pattern



Bir mülakat sorusu olabilir. ⇒ Switch case ile if arasındaki fark nedir?

Switch case sadece eşitliğe bakarken C# 9.0 yapısı ile gelen Relational patterns ile diğer karşılaştırmaları da yapabilemkteyiz. if yapısı tüm karşılaştırma ve eşitliklerde kullanılır

## ▼ Logical Pattern

And or ve not gibi mantıksal ifadeler kullanılabilemktedir.

# ▼ Not pattern

Not operatörü ile kullanılır.

```
string GetProduct(IProduct p) => p switch
{
    Technologic => "Teknolojik",
    Computer => "Bilgisayar",
    not Goggles => "Gözlük"
};

Not

Object obj = new Goggles();
if (obj is not Technologic)
{
    //...
}
not operatörünün
kullanılabildiği bir desendir.
```